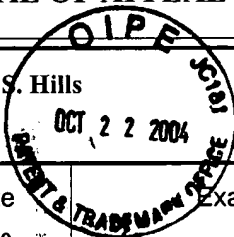


## TRANSMITTAL OF APPEAL BRIEF (Small Entity)

Docket No.  
TAJ-0002

In Re Application Of: Theodore S. Hills



Application No.

09/741,201

Filing Date

12/19/2000

Examiner

Lawrence J. Shrader

Customer No.

23413

Group Art Unit

2124

Confirmation No.

7649

Invention: METHOD FOR ENABLING A COMPILER OR INTERPRETER TO USE IDENTIFIERS FOUND AT RUN TIME IN A MAP CONTAINER OBJECT IN A MANNER SIMILAR OR IDENTICAL TO IDENTIFIERS DECLARED AT COMPILE TIME

COMMISSIONER FOR PATENTS:

Transmitted herewith in triplicate is the Appeal Brief in this application, with respect to the Notice of Appeal filed on:

☒ Applicant claims small entity status. See 37 CFR 1.27

The fee for filing this Appeal Brief is: \$170.00

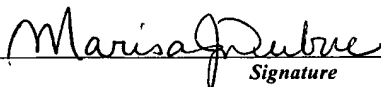
☒ A check in the amount of the fee is enclosed.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. 06-1130

☐ Payment by credit card. Form PTO-2038 is attached.

**WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.**

  
Signature

Dated: October 20, 2004

Marisa J. Dubuc  
Registration Number 46,673  
Cantor Colburn LLP  
55 Griffin Road South  
Bloomfield, CT 06002  
(860) 286-2929 phone  
(860) 286-0115 fax  
Customer Number 23413

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on  
October 20, 2004

(Date)



Signature of Person Mailing Correspondence

Alexis Zaruba

Typed or Printed Name of Person Mailing Correspondence

10/25/2004 HGUTEMA1 00000005 09741201

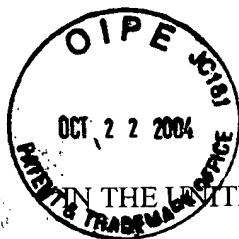
01 11:2402

170.00 00

cc:

AT 12/24 FEW

<b>CERTIFICATE OF MAILING BY FIRST CLASS MAIL (37 CFR 1.8)</b>			Docket No. TAJ-0002	
Applicant(s): Theodore S. Hills				
Application No. 09/741,201	Filing Date 12/19/2000	Examiner Lawrence J. Shrader	Customer No. 23413	Group Art Unit 2124
Invention: <b>METHOD FOR ENABLING A COMPILER OR INTERPRETER TO USE IDENTIFIERS FOUND AT RUN TIME IN A MAP CONTAINER OBJECT IN A MANNER SIMILAR OR IDENTICAL TO IDENTIFIERS DECLARED AT COMPILE TIME</b>				
<p>Transmittal of Appeal Brief (1 page); Appeal Brief (12 pages); 2 copies of Appeal Brief; Check No. 43196; Return Receipt Postcard</p> <p>I hereby certify that this _____ (Identify type of correspondence)</p> <p>is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on <u>October 20, 2004</u> (Date)</p> <p>Alexis Zaruba (Typed or Printed Name of Person Mailing Correspondence)</p> <p><u>Alexis Zaruba</u> (Signature of Person Mailing Correspondence)</p> <p>Note: Each paper must have its own certificate of mailing.</p> <div style="border: 1px solid black; height: 250px; width: 100%;"></div>				



APPELLANT: THEODORE S. HILLS )  
 ) Group Art Unit:  
 ) 2124  
SERIAL NUMBER: 09/741,201 )  
 ) Before the Examiner:  
FILED: December 19, 2000 ) Shrader, Lawrence J.  
 )  
FOR: METHOD FOR ENABLING A COMPILER )  
OR INTERPRETER TO USE )  
IDENTIFIERS FOUND AT RUN TIME IN )  
A MAP CONTAINER OBJECT IN A )  
MANNER SIMILAR OR IDENTICAL TO )  
IDENTIFIERS DECLARED AT COMPILE )  
TIME )

Mail Stop Appeal Brief – Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

### **APPEAL BRIEF**

#### **1. THE REAL PARTY IN INTEREST**

The real party in interest in this appeal is Tajea Corporation. Ownership by Tajea Corporation is established by assignment document recorded for this application on December 19, 2002 on Reel 011401, Frame 0448.

#### **2. RELATED APPEALS AND INTERFERENCES**

Appellant knows of no related patent applications or patents under appeal or interference proceeding.

#### **3. STATUS OF CLAIMS**

Claims 1-30 were originally filed in the above-referenced application. Of these, claims 6-10, 16-20, and 26-30 have been cancelled by the Appellant. Presently, claims 1-5, 11-15, and 21-25 remain pending. All pending claims stand rejected

TAJ-0002

under 35 U.S.C. §103(a). The Appellant hereby appeals the rejections of claims 1-5, 11-15, and 21-25.

4. STATUS OF AMENDMENTS

There have been no amendments filed subsequent to receipt of the final office action.

5. SUMMARY OF INVENTION

The following is a concise explanation of the invention. Reference to the specification and drawings is made pursuant to 37 CFR 41.37 and is not intended to limit the claims to the embodiments shown and described in the application.

Referring to the application, the invention relates to a mechanism for providing access to objects and their names defined in non-programming namespaces, and which is uniform across disparate kinds of namespaces. The invention enables a compiler or interpreter to use identifiers found at run time in a map container object in a manner similar to or identical to identifiers declared at compile time. The names defined in a map container object during the execution of a program can be treated in a manner similar to identical to names defined in a programming namespace at compilation or interpretation time.

Upon encountering a locally scoped identifier (represented as ‘a’ in Table 7, page 16 of Specification; Figure), the invention enables a compiler or interpreter to apply static name resolution rules as defined in the programming language definition (Table 6, page 15; Figure). If it does not resolve the identifier statically, and one or more map object container (represented as “Root” in Table 7, page 16; Figure) objects (e.g., ‘q’ in Table 7, page 16; Figure) have been incorporated into the current scope through “namespace using directives”, the compiler generates code as follows. First, the compiler regards the identifier as a reference to an object which is to be discovered at execution time. When generating object code for source referencing the identifier, the compiler generates code that accesses the referenced object indirectly, through a reference object associated with the identifier, whose value can be set at run time. Second, the compiler generates code to search at run time the map containers incorporated into the current scope through “namespace using directives” for a key

whose value is a string equal to the identifier string. If the generated code finds a matching key in the container, it sets the reference variable associated with the identifier to reference the object corresponding to the key found. If a match is not found, an exception is thrown. After successful assignment to the reference variable, the identifier may be used to reference the object found through its associated reference variable, just as if the identified object had been known at compile time (Tables 7-11; Figure1).

## 6. GROUNDS OF REJECTION

Claims 1, 2, 11, 12, 21, and 22 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,356,913 to Chu et al. (“Chu”) in view of U.S. Patent No. 5,692,180 to Lee et al. (“Lee”), and in further view of U.S. Patent No. 5,889,995 to Segnan. Claims 3, 4, 13, 14, 23, and 24 stand rejected under U.S.C. §103(a) as being unpatentable over Chu in view of Lee, and in further view of U.S. Patent No. 5,991,802 to Allard et al. (“Allard”). Claims 5, 15, and 25 stand rejected under U.S.C. §103(a) as being unpatentable over Chu in view of Lee, and in further view of U.S. Patent No. 6,173,313 to Klots et al. (“Klots”).

**A. Rejections under 35 U.S.C. §103(a) regarding Chu, Lee, and Segnan**

The Examiner improperly rejected claims 1, 2, 11, 12, 21, and 22 under 35 U.S.C. §103(a) as being unpatentable over Chu in view of Lee, and in further view of Segnan. For an obviousness rejection to be proper, the Examiner must meet the burden of establishing that all elements of the invention are disclosed in the prior art; and that the prior art relied upon, coupled with knowledge generally available in the art at the time of the invention, must contain some suggestion or incentive that would have motivated the skilled artisan to modify a reference or combined references. *In re Fine*, 5 U.S.P.Q.2d 1596, 1598 (Fed. Cir. 1988); *In Re Wilson*, 165 U.S.P.Q. 494, 496 (C.C.P.A. 1970); *Amgen v. Chugai Pharmaceuticals Co.*, 927 U.S.P.Q.2d, 1016, 1023 (Fed. Cir. 1996).

None of Chu, Lee, and Segnan teaches or suggests all of the elements set forth in claims 1, 2, 11, 12, 21, and 22. Claims 1, 11, and 21 recite, respectively, a method,

storage medium, and propagated signal for interpreting a source code comprising: “defining map container objects having keys, said keys comprising strings conforming to requirements for identifiers; using said keys of said map container objects as ordinary identifiers in the source code, said keys identifying objects found as value objects in said map container objects; using said map container objects as namespaces; declaring values of said keys as names qualified by said map container objects; and generating executable code to locate objects that are identified by said keys, wherein said executable code is executed after said compiling or interpreting said source code.”

The Examiner states that Chu discloses defining map container objects having keys...[the keys] comprising strings conforming to requirements for identifiers [and] using said keys of said map container objects as ordinary identifiers in the source code...said keys identifying objects found as value objects in said map container objects. (Office Action of April 20, 2004, pages 2-3). The Appellant respectfully disagrees. The identifier as recited in claims 1, 11, and 21 refers to an ordinary identifier, in that it is a string of characters formed according to the lexical requirements for identifiers in the programming language being processed. This identifier is found in the source code that is the input to the compiler. Chu makes no distinction between an “identifier” that is a number generated by a compiler for referencing objects, and an “identifier” that is an alphanumeric symbol that is, in turn, part of source code input to a compiler. An identifier that is an alphanumeric symbol is chosen by a programmer to present some concept in a program, usually mnemonically. By losing this distinction, Chu loses one of the values provided by Appellant’s claims 1, 11, and 21, which is that programmer-chosen symbols, not machine-generated numbers, can be used to identify objects, whether those objects are known at compile time or only discovered at run time.

The Examiner concedes that Chu does not recite “using said map container objects as namespaces” but asserts that Lee discloses using map container objects as namespaces to organize the objects (Office Action of April 20, 2004, page 3). The Examiner indicates that “it would have been obvious to one skilled in the art at the time the invention was made to combine the teaching of Chu regarding defining map container objects having keys, with the teaching of Lee wherein the container objects

are used as namespaces, because the said combination allows the facilitation of a directory reference, as taught by Lee” (Office Action of April 20, 2004, page 3). The Appellant respectfully disagrees with the Examiner’s statement. Claims 1, 11, and 21 do not teach generating an instruction that is a reference to a name space as suggested by the Examiner. Rather, claims 1, 11, and 21 teach the compiler generating code that accesses the referenced object indirectly through a reference object associated with the identifier whose value can be set “after said compiling or interpreting” (i.e., at run time). Also, the compiler generates code to search at run time the map containers incorporated into the current scope through namespace using directives. In other words, claims 1, 11, and 21 teach generating multiple instructions which will, at run time, search a namespace, not merely reference it, for an identifier. Moreover, the identifier is an ordinary identifier, in that it is a string of characters formed according to the lexical requirements for identifiers in the programming language being processed. This identifier is found in the source code that is the input to the compiler. Accordingly, the Appellant submits that Lee has been misapplied with respect to claims 1, 11, and 21.

The Examiner further argues that the combination of Chu, Lee, and Segnan teach “generating executable code to locate objects that are identified by said keys, wherein said executable code is executed after said compiling or interpreting said source code.” The Appellant strongly disagrees. In combination, the cited references might teach an ordinary identifier in the source code is converted to an integer (referred to in Segnan as an ‘identifier’) by a compiler, and then the integer is used to locate an object (referred to in Segnan as a ‘method’) at run time. This modification (combination), however, does not result in the Appellant’s teachings as recited in claims 1, 11, and 21. As claimed, the Appellant’s claims 1, 11, and 21 teach that the objects to be located are identified “after said compiling or interpreting said source code” (i.e., at run time) with character strings formed according to the requirements for ordinary identifiers, and are not identified with integers.

Additionally, an ordinary identifier is used in the source code to locate an object in a map container. An intermediate stage of converting the ordinary identifier to an integer as suggested by the combined references fails to achieve this element. The teachings of claims 1, 11, and 21 relate to and revolve around the use of the same

identifiers both at compilation time, as a static part of the source code being compiled, and at run time to search map containers. This carry-through of the identifier from compilation time to run time is not accomplished by the combination as suggested by the Examiner. Accordingly, the Appellant submits that the combination of Chu, Lee, and Segnan would not result in the teachings as recited in Appellant's claims 1, 11, and 21. The Appellant submits that claims 1, 11, and 21 are patentable over the references for at least these reasons.

Claim 2 depends from claim 1. Therefore, claim 2 includes all of the limitations set forth in claim 1. Claim 12 depends from claim 11. Therefore, claim 12 includes all of the limitations set forth in claim 11. Claim 22 depends from claim 21. Therefore, claim 22 includes all of the limitations set forth in claim 21. As explained above, Chu, Lee, and Segnan do not teach or suggest all of the limitations of claims 1, 11, and 21. Accordingly, the Appellant submits that the rejections of claims 2, 12, and 22 under 35 U.S.C. §103(a) is improper.

#### **B. Rejections under 35 U.S.C. §103(a) regarding Chu, Lee, and Allard**

The Examiner improperly rejected claims 3, 4, 13, 14, 23, and 24 under 35 U.S.C. §103(a) as being unpatentable over Chu in view of Lee, and in further view of Allard. As stated previously, for an obviousness rejection to be proper, the Examiner must meet the burden of establishing that all elements of the invention are disclosed in the prior art; and that the prior art relied upon, coupled with knowledge generally available in the art at the time of the invention, must contain some suggestion or incentive that would have motivated the skilled artisan to modify a reference or combined references. *In re Fine*, 5 U.S.P.Q.2d 1596, 1598 (Fed. Cir. 1988); *In Re Wilson*, 165 U.S.P.Q. 494, 496 (C.C.P.A. 1970); *Amgen v. Chugai Pharmaceuticals Co.*, 927 U.S.P.Q.2d, 1016, 1023 (Fed. Cir. 1996).

Claims 3 and 4 depend from claim 1. Therefore, claims 3 and 4 include all of the limitations set forth in claim 1. Claims 13 and 14 depend from claim 11. Therefore, claims 13 and 14 include all of the limitations set forth in claim 11.



Claims 23 and 24 depend from claim 21. Therefore, claims 23 and 24 include all of the limitations set forth in claim 21. As explained above, Chu, Lee, and Segnan do not teach or suggest all of the limitations of claims 1, 11, and 21. Accordingly, the Appellants submit that the rejections of claims 3, 4, 13, 14, 23, and 24 under 35 U.S.C. §103(a) is improper.

Notwithstanding, with respect to Appellant's claims 3, 13, and 23, the Examiner states that Allard teaches "...said objects are identified by Uniform Resource Identifiers (URIs)" (Office Action of April 20, 2004, page 7). The Appellant disagrees and submits that the Allard reference is directed to Uniform Resource Locators (URLs), rather than Uniform Resource Identifiers (URIs) as recited in claims 3, 13, and 23. Thus, the Appellant submits that Allard has been misapplied to claims 3, 13, and 23.

### **C. Rejections under 35 U.S.C. §103(a) regarding Chu, Lee, and Klots**

The Examiner improperly rejected claims 5, 15, and 25 under 35 U.S.C. §103(a) as being unpatentable over Chu in view of Lee, and in further view of Klots. This grouping of claims is proper as the claims contain distinguishing features that are patentable apart from the independent base claims from which they depend. As stated previously, for an obviousness rejection to be proper, the Examiner must meet the burden of establishing that all elements of the invention are disclosed in the prior art; and that the prior art relied upon, coupled with knowledge generally available in the art at the time of the invention, must contain some suggestion or incentive that would have motivated the skilled artisan to modify a reference or combined references. *In re Fine*, 5 U.S.P.Q.2d 1596, 1598 (Fed. Cir. 1988); *In Re Wilson*, 165 U.S.P.Q. 494, 496 (C.C.P.A. 1970); *Amgen v. Chugai Pharmaceuticals Co.*, 927 U.S.P.Q.2d, 1016, 1023 (Fed. Cir. 1996).

Claim 5 depends from claim 1. Therefore, claim 5 includes all of the limitations set forth in claim 1. Claim 15 depends from claim 11. Therefore, claim 15 includes all of the limitations set forth in claim 11. Claim 25 depends from claim 21. Therefore, claim 25 includes all of the limitations set forth in claim 21. As explained above, Chu, Lee, and Segnan do not teach or suggest all of the limitations of claims 1,

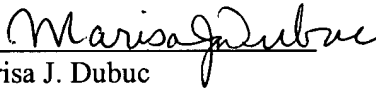
11, and 21. Accordingly, the Appellant submits that the rejection of claims 5, 15, and 25 under 35 U.S.C. §103(a) is improper.

**A. Conclusion**

For the reasons cited above, Appellant respectfully submits that the rejections are improper and request reversal of the outstanding rejections. If there are any additional charges with respect to this Appeal, or otherwise, please charge them to Deposit Account No. 06-1130 maintained by Appellant's attorneys.

Respectfully submitted,

THEODORE S. HILLS

By:   
Marisa J. Dubuc  
Registration No. 46,673  
CANTOR COLBURN LLP  
55 Griffin Road South  
Bloomfield, CT 06002  
Telephone (860) 286-2929  
Facsimile (860) 286-0115  
Customer No. 23413

Date: October 20, 2004

## APPENDIX A

### *Appealed Claims*

Claim 1. A method of compiling or interpreting a source code comprising:  
    defining map container objects having keys, said keys comprising strings conforming to requirements for identifiers;  
    using said keys of said map container objects as ordinary identifiers in the source code, said keys identifying objects found as value objects in said map container objects;  
    using said map container objects as namespaces;  
    declaring values of said keys as names qualified by said map container objects; and  
    generating executable code to locate objects that are identified by said keys, wherein said executable code is executed after said compiling or interpreting said source code.

Claim 2. The method of claim 1 wherein said objects are in a filesystem.

Claim 3. The method of claim 1 wherein said objects are identified by Uniform Resource Identifiers (URIs).

Claim 4. The method of claim 1 wherein said objects are identified by environment variables.

Claim 5. The method of claim 1 wherein said objects are identified by run-time object name resolution schemes.

Claim 11. A storage medium encoded with machine-readable code, the code including instructions for causing a computer to implement a method of compiling or interpreting a source code comprising:

defining map container objects having keys, said keys comprising strings conforming to requirements for identifiers;

using said keys of said map container objects as ordinary identifiers in the source code, said keys identifying objects found as value objects in said map container objects;

using said map container objects as namespaces;

declaring values of said keys as names qualified by said map container objects; and

generating executable code to locate objects that are identified by said keys, wherein said executable code is executed after said compiling or interpreting said source code.

Claim 12. The storage medium of claim 11 wherein said objects are in a filesystem.

Claim 13. The storage medium of claim 11 wherein said objects are identified by Uniform Resource Identifiers (URIs).

Claim 14. The storage medium of claim 11 wherein said objects are identified by environment variables.

Claim 15. The storage medium of claim 11 wherein said objects are identified by run-time object name resolution schemes.

Claim 21. A signal propagated over a propagation medium, the signal encoded with code, the code including instructions for causing a computer to implement a method of compiling or interpreting a source code comprising:

defining map container objects having keys, said keys comprising strings conforming to requirements for identifiers;

using said keys of said map container objects as ordinary identifiers in the source code, said keys identifying objects found as value objects in said map container objects;

using said map container objects as namespaces;

declaring values of said keys as names qualified by said map container objects; and

generating executable code to locate objects that are identified by said keys, wherein said executable code is executed after said compiling or interpreting said source code.

Claim 22. The signal propagated over the propagation medium of claim 21 wherein said objects are in a filesystem.

Claim 23. The signal propagated over the propagation medium of claim 21 wherein said objects are identified by Uniform Resource Identifiers (URIs).

Claim 24. The signal propagated over the propagation medium of claim 21 wherein said objects are identified by environment variables.

Claim 25. The signal propagated over the propagation medium of claim 21 wherein said objects are identified by run-time object name resolution schemes.